EURASIP Journal on Applied Signal Processing 2004:2, 290–303

Error Resilient Video Compression Using Behavior Models

Jacco R. Taal
Information and Communication Theory Group, Department of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
Email: j.r.taal@ewi.tudelft.nl

Zhibo Chen
Imaging Technology Group, IMNC, Sony Corporation, 6-7-35 Kitashinagawa, Shinagawa-Ku, Tokyo 141-0001, Japan
Email: chenzhibo@tsinghua.org.cn

Yun He
Video Communication Research Group, Electronic Engineering Department, Tsinghua University, 11-425 East Main Building, 100084 Beijing, China
Email: hey@video.mdc.tsinghua.edu.cn

R. (Inald) L. Lagendijk
Information and Communication Theory Group, Department of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
Email: r.l.lagendijk@ewi.tudelft.nl

Received 1 December 2002; Revised 26 September 2003

Wireless and Internet video applications are inherently subjected to bit errors and packet errors, respectively. This is especially so if constraints on the end-to-end compression and transmission latencies are imposed. Therefore, it is necessary to develop methods to optimize the video compression parameters and the rate allocation of these applications that take into account residual channel bit errors. In this paper, we study the behavior of a predictive (interframe) video encoder and model the encoders behavior using only the statistics of the original input data and of the underlying channel prone to bit errors. The resulting data-driven behavior models are then used to carry out group-of-pictures partitioning and to control the rate of the video encoder in such a way that the overall quality of the decoded video with compression and channel errors is optimized.

Keywords and phrases: behavior model, rate distortion, video coding, error resilience.

1. INTRODUCTION

Although the current video compression techniques can be considered mature, there are still many challenges in the design and operational control of compression techniques for end-to-end quality optimization. This is in particular true in the context of unreliable transmission media such as the Internet and wireless links. Conventional compression techniques such as JPEG and MPEG were designed with error-free transmission of the compressed bitstream in mind. With such unreliable media, not all bit or packet errors may be corrected by retransmissions or forward error correction (FEC). Depending on the kind of channel coder, residual channel errors may be present in the bitstream after channel decoding. In most practical packet network systems, packet retransmission corrects for some, but not all, packet losses. Classic rate control, such as TM.5 in MPEG [1], can be used to control the video encoder according to the available bit rate offered by the channel coder; adaptation to the bit error rate by inserting intracoded blocks is nevertheless not incorporated in TM.5. Other methods that control the insertion of intracoded blocks exist [2].

Three classes of error resilient source coding techniques that deal with error prone transmission channels may be distinguished. The first well-known approach is joint source-channel coding, which aims to intimate integration of the source and channel coding algorithms [3, 4]. Although this intimate integration brings several advantages to the end-to-

end quality optimization, it comes at the price of a significant complexity increase. Furthermore, nearly all of these approaches only work with specific or nonstandard network protocols and with a specific video encoder and/or decoder.

The second class represents many approaches where the source coder has no (or limited) control of the network layer. It is important to understand that these approaches can not be generally optimal since the channel coder and the source coder are not jointly optimized. Since there is no joint optimization, the only thing the source coder can do is to adapt its own settings according to the current behavior of the network layer. In many applications, joint optimization is impossible because none of the standard network protocols (IP, TCP, and UDP) support this. Even though the source coder has no or limited control over the network layer, the rate control algorithm can adapt to the available bit rate and to the amount of residual bit errors or packet losses. Such a control algorithm needs a model describing the effects of bit errors or packet losses on the overall distortion.

The third class contains the approaches advocated in [5, 6]. In these approaches, the best properties of the first two classes are combined. Here, the authors propose to limit the integration to joint parameter optimization, so that there is no algorithmic integration. In previous work at Delft University of Technology [7], an efficient overall framework was proposed for such joint parameter optimization from a quality-of-Service (QoS) perspective. This framework requires high-level and abstract models describing the behavior of source and channel coding modules. However, this framework had not yet been tested with a real video coder and with a real behavior model.

In this paper, we propose such a behavior model for describing source-coding characteristics, giving some information about the channel coder. Although this model is designed to be used in a QoS setup, it may also be used to optimize the encoders settings when we only have knowledge of, but no control over, the current channel (as a second class approach).

With this behavior model, we can predict the behavior of a source coder in terms of the image quality related to the channel coder parameters: the bit rate, the bit error rate (BER), and the latency. To be applicable in a real-time and perhaps low power setup, the model itself should have a low complexity and should not require that many frames have to reside in a buffer (low latency).

We evaluate the behavior models with one type of progressive video coder. However, we believe that other coders can be described fairly easily with our methods as well, since we try to describe the encoders at the level of behavior rather than at a detailed algorithmic or implementation level. In Section 2, we first discuss our combined source-channel coding system; the problem we wish to solve, and we describe the source and channel coders on a fairly high abstraction level. From these models, we can formulate the end-to-end quality control as an optimization problem, which we will discuss in Section 3. Section 4 describes in depth the construction of the proposed models. In Section 5, our models are validated in a simulation where a whole group of pictures (GOP) were transmitted over an error prone channel. Section 6 concludes this paper with a discussion.

2. PROBLEM FORMULATION

To optimize the end-to-end quality of compressed video transmission, one needs to understand the individual components of the link. This understanding involves knowledge of the rate distortion performance and the error resilience of the video codec, of the error correcting capabilities of the channel codec, and possibly of parameters such as delay, jitter, and power consumption. One of the main challenges in attaining an optimized overall end-to-end quality is the determination of the influence of the individual parameters controlling the various components. Especially because the performances of various components depend on each other, and the control of these parameters is not straightforward.

In [4, 5, 8, 9], extensive analyses of the interaction and trade-offs between source and channel coding parameters can be found. A trend in these approaches is that the underlying components are modeled at a fairly high abstraction level. The models are certainly independent of the actual hardware or software implementation but they also become more and more independent of the actual compression or source coding algorithm used. This is in strong contrast to the abundance of joint source channel coding approaches, which typically optimize a particular combination of source and channel coders, utilizing specific internal algorithmic structures and parameter dependencies. Although these approaches have the potential to lead to the best performance, their advantages are inherently limited to the particular combination of coders and to the conditions (source and channel) under which the optimization was carried out.

In this paper, we refrain from the full integration of source and channel codecs (i.e., the joint source-channel coding approach) but we keep the source and channel coders as much separate as possible.

The interaction between source and channel coders and, in particular, the communication of key parameters is encapsulated in a QoS framework. The objective of the QoS framework is to structure the communication context parameters between OSI layers. In the scope of this paper, the context can be defined not only by radio/Internet channel conditions, but also by the demands of the application or device concerning the quality or the complexity of the video encoding. Here we discuss only the main outline of the QoS interface. A more detailed description of the interface can be found in the literature (see [6, 7]).

Figure 1 illustrates the QoS Interface concept [7]. The source and channel coders operate independent of each other, but are both under the control of QoS controllers. The source coder encodes the video data, thereby reducing the needed bit rate. The channel coder protects this data. It decreases the BER, thereby effectively reducing the bit rate available for source coding and increasing the latency. The QoS controller of the source coder communicates the key parameters—in this case, the bit rate, the BER, and latency—

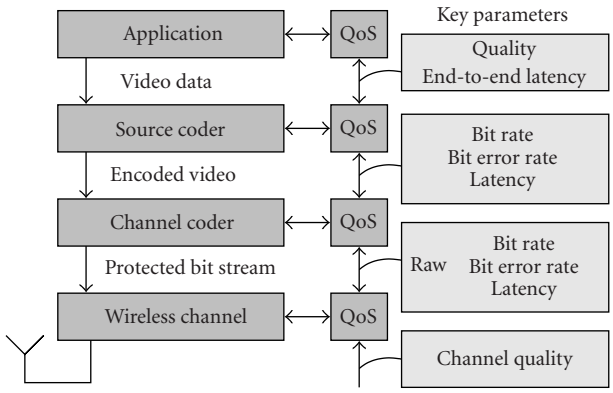

FIGURE 1: QoS concept: different (OSI)layers are not only communicating their payloads, they are also controlled by QoS controllers that mutually negotiate to optimize the overall performance.

with the QoS controller of the channel coder. Based on the behavior description of the source and channel coding modules, the values of these parameters are optimized by the QoS controller. In a practical system, this optimization takes into account context information about the application (e.g., maximum latency) and about the channel (e.g., throughput at the physical layer). The application may set constraints on the operation of the lower layers, for instance, on the power consumption or the delay. In this paper, we assume that the only constraint set by the application is the end-to-end delay T_a.

In order to implement the QoS Interface/controller concept, the following three problems need to be solved.

(i) The key parameters must be optimized over different (OSI) layers. We have developed the "adaptive resource contracts" (ARC) approach for solving this problem. ARC exchanges the key parameters between two layers such that after a negotiation phase, both layers agree on the values of these parameters. These key parameters represent the trade-offs that both layers have made to come to a joint solution of the optimization. A detailed discussion of ARC falls outside the scope of this paper. We refer to [6, 7, 10].

(ii) The behavior of the source and channel coders should be modeled parametrically such that joint optimization of the key parameters can take place. At the same time, an internal controller that optimizes the performance of the source and channel coders independently, given the already jointly optimized key parameters, should be available. The emphasis in this paper is on the modeling of the video coder behavior.

(iii) An optimization procedure should be designed for selecting the parameters internal to the video codec, given the behavior model and the key parameters. We do not emphasize this aspect of the QoS interface in this paper as we believe that the required optimization procedure can be based on related work as that in [11].

In previous work and analyses [6, 7], the source coder was modeled as a *progressive encoder*, which means that with every additionally transmitted bit, the quality of the received decoded information increases. Therefore, the most important information is encoded at the beginning of the data stream, and the least important information is encoded at the end. In principle, we believe that any progressive encoder can be described with our models. To keep things simple from a compression point of view, we use the common interframe coding structure (with one interframe per GOP, multiple predictively encoded interframes, and no bidirectional encoded frames). The actual encoding of the (difference) frames is done by a JPEG2000 (see [12]) encoder, which suits our demand for progressive behavior. Figure 2 shows the typical block diagram of this JPEG2000-based interframe coder. In this paper, we exclude motion compensation of interframes for simplicity reasons. The internal parameters for this video encoder are the number of frames in a GOP N, and the bit rates r_i for the individual frames. Symbols X_i and X_{i-1} denote the current frame and the previous frame, $\overline{X}$ denotes a decoded frame, and $\tilde{X}$ denotes a decoded frame at the decoder side, possibly with distortions caused by residual channel errors. Symbols $\tilde{D}_{\mathrm{q}}$ and $\tilde{D}_{\mathrm{e}}$ denote the quantization distortion and the distortions caused by residual channel errors (named "channel-induced distortion" hereafter), respectively.

In this work, the channel coder is defined as an abstract functional module with three interface parameters. The channel coder has knowledge of the current state of the channel which it is operating on. Therefore, it can optimize its own internal settings using behavior models. Such a channel coder may use different techniques like FEC and automatic repeat requests (ARQ) to protect the data at the expense of added bit rate and increased delay (latency). The exact implementation is nevertheless irrelevant for this paper. From here we will assume that the error protection is not perfect because of latency constraints; therefore the residual BER may be non zero. The behavior models can be obtained by straightforward analysis of the channel coding process [5].

3. SOURCE ENCODER OPTIMIZATION CRITERION

At this point, we assume that we have a behavior model for our video encoder. The development of this behavior model is the subject of Section 4. Given the behavior model, we can minimize the average end-to-end distortion $\hat{D}$ given the constraints imposed by the QoS interface. In our work, the QoS Interface negotiates three key parameters between source and channel coder, namely, $\{R, \mathrm{BER}, T_c\}$, with

(i) R: the available bit rate for source coding (average number of bits per pixel);

(ii) the residual BER: the average bit error rate after channel decoding;

(iii) T_c: the average time between handing a bit to the channel encoder, and receiving the same bit from the channel decoder.

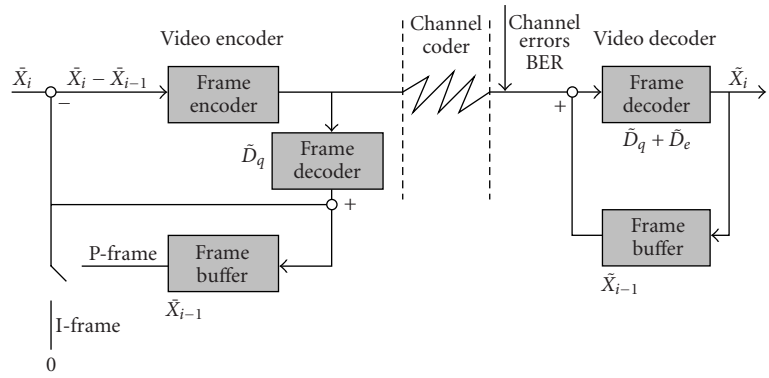

Figure 2: Simple video coding scheme. The "frame encoder" and "frame decoder" blocks represent the single frame encoder and decoder. The "frame buffer" is needed for the predictively encoded interframes.

The resulting source coding optimization problem now becomes the minimization of the distortion D, which can be formulated as follows:

$$\min_{I_{\text{src}}} D(I_{\text{src}} \mid \{R, \text{BER}, T_c\}). \tag{1}$$

Here, I_{src} denotes the set of internal source coder parameters over which the performance of the encoder must be optimized, given the key parameters $\{R, \text{BER}, T_c\}$. The actual set of internal parameters to be considered depends on the encoder under consideration and the parameters included in the encoders behavior model. In this paper, we consider the optimization of the following internal parameters:

(i) N, the length of the current GOP. Each GOP starts with an intraframe and is followed by $N-1$ predictively encoded interframes;

(ii) $\vec{r} = \{r_0, r_1, \ldots, r_{N-1}\}$: the target bit rate for each individual frame in a GOP.

The encoder parameter N relates to the coding efficiency and the robustness of the compressed bitstream against the remaining errors. The larger is N, the higher the coding efficiency, because more interframes are encoded. At the same time, the robustness of the stream is lower due to the propagation of decoded transmission errors.

On the other hand, in order to optimize the settings $\{N, \vec{r}\}$ for $N_{\max}$ frames, these $N_{\max}$ frames have to be buffered, thereby introducing a latency. In our approach, the QoS interface prescribes the maximum end-to-end latency T_a (seconds), and we assume that the channel coder will have an end-to-end latency of T_c (seconds), from the channel encoder to the channel decoder, including transmission. Analysis of the whole transmission chain gives the following expression for the total end-to-end latency:

$$T_a = \frac{N-1}{f_r} + T_e + T_c + \frac{B}{R}, \tag{2}$$

where f_r is the frame rate of the video sequence that is encoded and T_e is the upper bound of the time it takes to encode a frame. Finally B/R is the transmission time for one frame B/R; the maximal number of bits to describe a frame divided by the channel coding bit rate R.

We can now find an expression for the maximal number of frames that can be in the buffer while still meeting the end-to-end latency constraints T_a. Clearly B/R is only known after allocating the rate for each frame. We suggest taking the worst case value for B (i.e., calculated from the maximal bit rate setting). The same goes for T_E where we suggest to take the worst case encoding time per frame,

$$N_{\max} = 1 + \left(T_a - T_e - T_c - \frac{B}{R}\right) f_r. \tag{3}$$

In each frame i, two kinds of distortion are introduced: (1) the quantization error distortion, denoted by D_q and (2) the channel-induced distortion caused by bit errors in the received bitstream, denoted by D_e. With our optimization problem, we aim to minimize the average distortion, which is the sum of individual distortions of a GOP divided by the length of the group:

$$D_{\text{GOP}} = \frac{1}{N} \sum_{i=0}^{N-1} \{D_{\text{q}}(r_i) + D_{\text{e}}(r_i, \text{BER})\}. \tag{4}$$

Following [5], we assume that D_q and D_e within one frame are mutually independent. Although (4) is a simple additive distortion model, the distortion of a frame is still dependent on that of the previous frames because of the interframe prediction. Therefore, in our models, we have to take into account the propagation of quantization and channel-induced distortions.

Taking the above parameters into account, we can now rewrite (1) as the following bit rate allocation problem:

$$\begin{aligned} \vec{r}_{\text{opt}}, N_{\text{opt}} &\longleftarrow \min_{\vec{r},N} D_{\text{GOP}}(\vec{r}, N \mid \text{BER}) \\ &= \min_{N} \left\{ \min_{\vec{r}} \frac{1}{N} \sum_{i=0}^{N-1} D_{\text{q}}(r_i) + D_{\text{e}}(r_i, \text{BER}) \right\} \end{aligned} \tag{5}$$

subject to

$$\frac{1}{N}\sum_{i=1}^{N-1} r_i = R, \quad N \le N_{\max}. \tag{6}$$

The approach that we follow in this paper is to optimize the bit rate allocation problem (5) and (6) based on two frame-level parametric behavior models. The first (rate distortion) model parametrically describes the relation between the variance of the quantization distortion and the allocated bit rate based on the variance of the input frames. The second (channel-induced distortion) model parametrically describes the relation between the variance of the degradations due to transmission and the decoding errors based on the variance of the input frames and the *effective* (BER).[1]

4. RATE DISTORTION MODEL

In this section, we first propose a behavior model for the rate distortion characteristics D_q of video encoders and then propose a model for distortion caused by residual channel errors including the error propagation D_e.

There are two approaches for modeling the rate distortion (RD) behavior of sources. The first approach is the analytical approach, where mathematical relations are derived for the RD functions assuming certain (stochastic) properties of the source signal and the coding system. Since these assumptions do not often hold in practice, the mismatch between the predicted rate distortion and the actual rate distortion is (heuristically) compensated for by empirical estimation. The second is the empirical approach where the RD functions are modeled through regression analysis of the empirically obtained RD data. The rate distortion model proposed in [5] is an example of an empirical model of the distortion of an entire encoder for a given bit rate.

In our work, we anticipate the real-time usage of the constructed abstract behavior models. At the same time, we want to keep the complexity of the models low. This limits the amount of *preprocessing* or *analysis* that we may do on the frames to be encoded. Therefore, we will base our behavior models on variance information only. In particular, we will use

(i) the variance of the frame under consideration denoted by $\mathrm{VAR}[X_i]$,

(ii) the variance of the difference of two consecutive frames denoted by $\mathrm{VAR}[X_i - X_{i-1}]$.

4.1. Rate distortion behavior model of intraframes

It is well known that for memoryless Gaussian distributed sources X with variance $\mathrm{VAR}[X]$, the RD function is given by

$$r(D_q) = \frac{1}{2}\log_2\left(\frac{\mathrm{VAR}[X]}{D_q}\right), \tag{7}$$

or when we invert this function by

$$D_q(r) = \mathrm{VAR}[X]2^{-2r}. \tag{8}$$

Empirical observations show that for the most common audio and video signals under small distortions, the power function $-2r$ gives an accurate model for the behavior of a compression system especially in terms of the quality gain per additional bit (in bit rate terms) spent. For instance, the power function $-2r$ leads to the well-known result that, at a sufficiently high bit rate, for most video compression systems we gain approximately 6 dB per additional bit per sample.

However, for more complicated compression systems and especially for larger distortions, the simple power function does not give us enough flexibility to describe the empirically observed RD curves, which usually give more gain for the same increase in bit rate. Since there is basically no theory to rely on for these cases without extremely detailed modeling of the compression algorithm, we instead propose to generalize (8) as follows:

$$D_q(r) = \mathrm{VAR}[X]2^{f(r)}. \tag{9}$$

The function $f(r)$ gives us more freedom to model the (RD) behavior at the price of regression analysis or online parameter estimation on the basis of observed rate distortion realizations. The choice of the kind of the function used to model $f(r)$ is a pragmatic one. We have chosen a third-order polynomial function. A first- or second-order function was simply too imprecise, while a fourth-order model did not give a significant improvement and higher-order models would defeat our objective of finding simple and generic models. Clearly there is a trade-off between precision (high order) and generality (low order).

In Figure 3, we show the (RD) curve of the experimentally obtained $\tilde{D}_q(r)$ for the JPEG2000 compression of the first frame of the Carphone sequence for bit rates between 0.05 and 1.1 bits per pixel (bpp). The solid line represents a third-order polynomial fit of $f(r)$ on the measured values. This fit is much better than the linear function $f(r) = -2r$. The following function was obtained for the first frame of the Carphone sequence:

$$D_q(r) = \mathrm{VAR}[X]2^{-4.46r^3+11.5r^2-12.7r-1.83}. \tag{10}$$

It is interesting to see how the RD curve changes for different frames of the same scene or different scenes. Figure 4 shows the RD curve for frame 1 and frame 60 of Carphone, and frame 1 of Foreman. Observe that the Carphone frames have very similar curves. The Foreman curve is shifted, but is still similar to the other two. These observations strengthen our belief that the model is generally applicable for this type of coder. Of course the $f(r)$ needs to be fitted for a particular sequence, on the other hand, we believe that a default curve $f_0(r)$ can be used to bootstrap the estimation of model parameters for other video sequences. The function $f(r)$ can then be adapted with a new RD data as the encoding continues.

[1] By "effective bit error rate" we mean the residual bit error rate, that is, the bit errors that are still present in the bitstream after channel decoding.

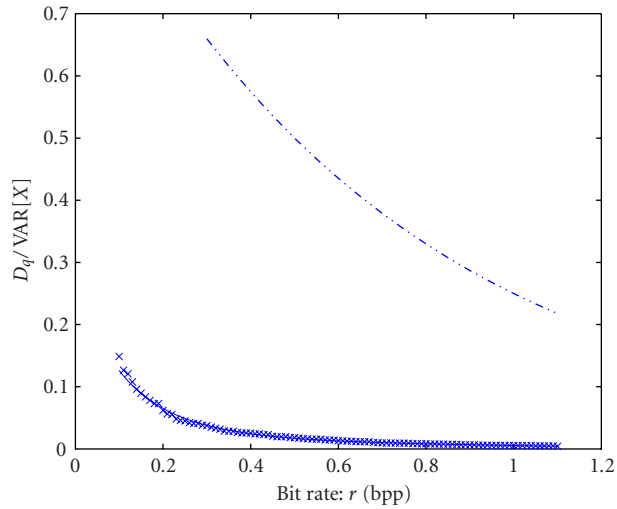

FIGURE 3: RD curve for the first frame in Carphone. The crosses (×) are the measured normalized distortions $\tilde{D}_q$ and the solid line corresponds to the fitted function $2^{f(r)}$. The dashed-dotted line corresponds to the RD model 2^{-2r}.

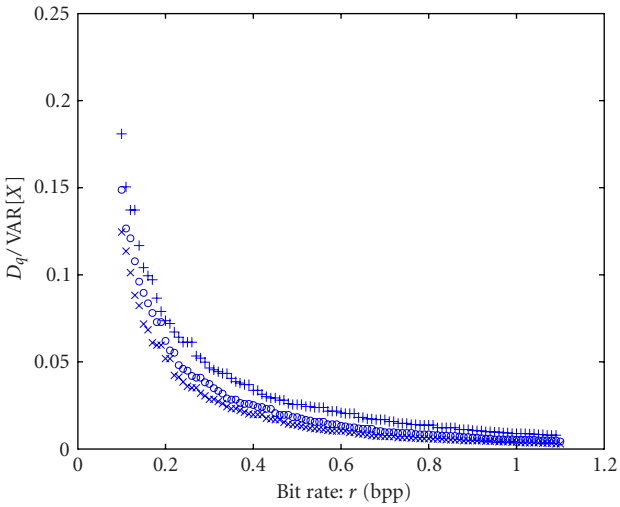

FIGURE 4: Intraframe RD curve for the first frame of Carphone (×), frame 60 of Carphone (∘), and the first frame of Foreman (+).

4.2. *Rate distortion behavior model of interframes*

For modeling the (RD) behavior of interframes, we propose to use a model similar to the one in (9), but with a different polynomial $g(r)$,

$$D_q(r_i) = \mathrm{VAR}\left[X_i - \overline{X}_{i-1}\right]2^{g(r_i)}. \tag{11}$$

Here, $\overline{X}_{i-1}$ denotes the previously decoded frame $i-1$, whereas with intraframes, a third-order polynomial was needed to predict $f(r)$ accurately enough. With interframes, a second-order polynomial was sufficient to predict $g(r)$. The reason for this can be found in the fact that interframes are less correlated than intraframes. Therefore, $g(r)$ is more similar to the theoretical $-2r$ than $f(r)$.

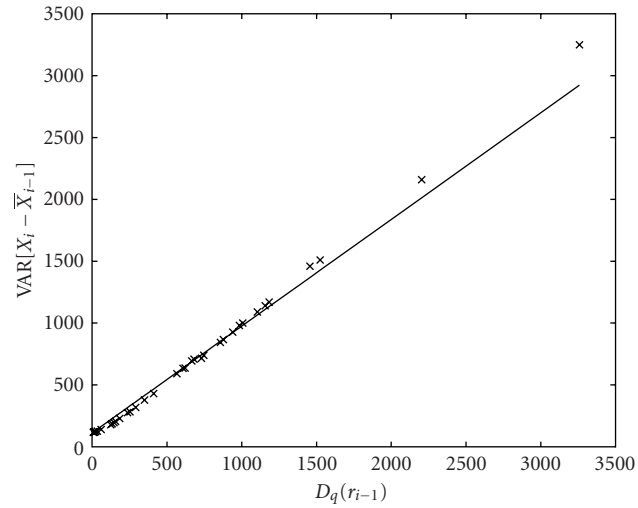

FIGURE 5: The relationship between the variance of frame difference $\mathrm{VAR}[X_i - \overline{X}_{i-1}]$ and the quantization distortion $\tilde{D}_q(r_{i-1})$. The fitted line describes $\mathrm{VAR}[X_i - \overline{X}_{i-1}] = \mathrm{VAR}[X_i - X_{i-1}] + \kappa D_q(r_{i-1})$.

In (11), $\mathrm{VAR}[X_i - \overline{X}_{i-1}]$ is the variance of the difference between the current frame i and the previously *encoded* frame $i-1$. Since the latter is only available *after* encoding (and thus after solving (5) and (6)), we need to approximate $\mathrm{VAR}[X_i - \overline{X}_{i-1}]$. Obviously we have

$$\begin{aligned}\mathrm{VAR}\left[X_i - \overline{X}_{i-1}\right] &= E\left[\left(X_i - \overline{X}_{i-1}\right)^2\right] \\ &= E\left[\left((X_i - X_{i-1}) - (\overline{X}_{i-1} - X_{i-1})\right)^2\right] \\ &= \mathrm{VAR}\left[X_i - X_{i-1}\right] + D_q(r_{i-1}) \\ &\quad - 2E\left[(X_i - X_{i-1})(\overline{X}_{i-1} - X_{i-1})\right].\end{aligned} \tag{12}$$

The last term on the right-hand side of (12) cannot be easily estimated beforehand and should therefore be approximated. We collapse this entire term into a quantity that only depends on the amount of quantization errors D_q from the previous frame, yielding

$$\mathrm{VAR}\left[X_i - \overline{X}_{i-1}\right] = \mathrm{VAR}\left[X_i - X_{i-1}\right] + \kappa D_q(r_{i-1}). \tag{13}$$

We expect the quantization noise of frame X_{i-1} to be only slightly correlated with the frame difference between frames X_{i-1} and X_i. Therefore, we expect the value of κ to be somewhat smaller than one. Note that by combining (13) and (11), D_q is defined recursively, thereby making (5) and (6) a dependent optimization problem.

Figure 5 illustrates the relation between the frame difference variance $\mathrm{VAR}[X_1 - \overline{X}_0]$ and the quantization distortion of the first frame of Carphone $\tilde{D}_q$. The first frame is encoded at different bit rates. We observe a roughly linear relation, in this case, with an approximate value of $\kappa = 0.86$.

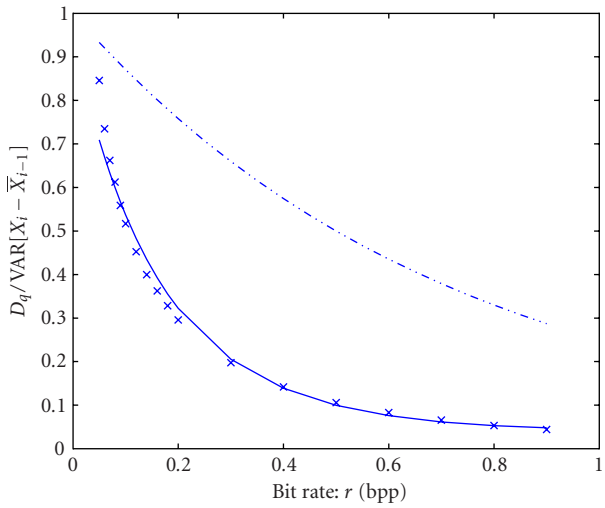

Figure 6: Average RD curve for the first interframe of Carphone. The crosses (×) are the measured normalized distortions $\tilde{D}_q(r_i)$ and the solid line corresponds to the fitted function $2^{g(r)}$. The dashed-dotted line corresponds the RD model 2^{-2r}.

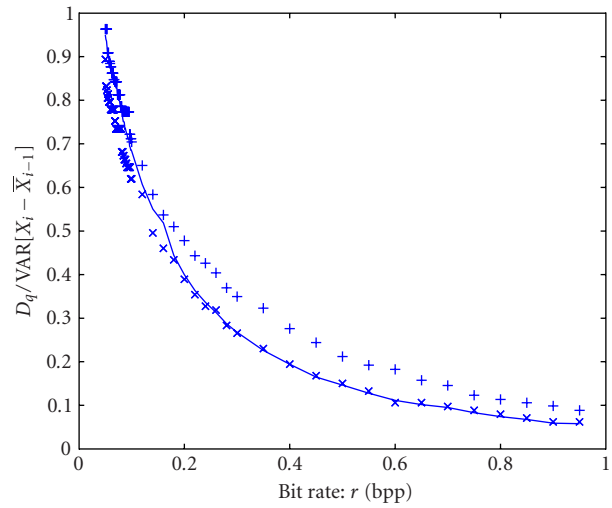

Figure 7: RD curve for the first interframe of Carphone (×), the average RD curve for the first ten frames of Carphone (—), and the RD curve for the first frame of Foreman (+).

We observed similar behavior for other sequences such as Susie and Foreman as well. We therefore postulate that (13) is an acceptable model for calculating the variance $\mathrm{VAR}[X_i - \overline{X}_{i-1}]$ as needed in (11).

The variance $X_i - \overline{X}_{i-1}$ consists of two terms: the quantization distortion of the previous frames, and the frame difference between the current and the previous frame. These two terms might show different RD behavior, that is, a separate $g(r)$ for both terms. However, we assume that both signals show the same behavior since they are both frame-difference signals by nature and not whole frames. The model for predicting the distortion of an interframe now becomes

$$D_q(r_i) = \left(\mathrm{VAR}\left[X_i - X_{i-1}\right] + \kappa D_q(r_{i-1})\right) 2^{g(r_i)}. \tag{14}$$

Figure 6 shows the experimentally obtained RD curve together with a fitted curve representing our model (14). Since this RD curve should not only be valid for varying bit rate r_i but also for varying propagated quantization distortion $D_q(r_{i-1})$, we also vary the bit rate of the previous frame r_{i-1}. Both rates were varied from 0.05 to 0.9. Each value of $D_q(r_i)$ is an average over all settings of r_{i-1}. For completeness, the theoretic curve (8) is shown as well. The function that describes the RD behavior for these frames is

$$D_q(r_i) = \left(\mathrm{VAR}\left[X_i - X_{i-1}\right] + \kappa D_q(r_{i-1})\right) 2^{3.86r_i^2 - 8.15r_i - 0.26}. \tag{15}$$

We then compare the curves for different frames. Figure 7 shows the RD curve for the first frame difference of Carphone and the RD curve for the first frame difference of Foreman as well as the average RD curve for the first ten frame differences of Carphone. This shows again that these curves do not vary much for different video frames and different video sources.

4.3. Channel-induced distortion behavior model

When the channel suffers from high error rates, the channel decoding will not be able to correct all bit errors. Therefore, to solve (5) and (6), we also need a model that describes the behavior of the video decoder in the presence of bitstream errors.

First, we define the channel-induced distortion to be the variance of the difference between the decoded frame ($\overline{X}$) at the encoder side and the decoded frame at the decoder side ($\tilde{X}$):

$$\tilde{D}_e = \mathrm{VAR}[\tilde{X} - \overline{X}]. \tag{16}$$

In [9], a model that describes the coders vulnerability to packet error losses is proposed:

$$D_e = \sigma_{u_0}^2 \,\mathrm{PER}, \tag{17}$$

where $\sigma_{u_0}^2$ is an empirical constant and is found empirically and PER is the packet error rate. Since we are dealing with bit errors and want to predict the impairment on a frame-per-frame basis, we look for a better model.

Modeling the impairments that are due to uncorrected bit errors may result in a detailed analysis of the compression technique used (see, e.g., [13]). Since we desire to have an abstract and a high level model with a limited number of parameters, we base our model on the following three empirical observations.

(1) For both intraframes and interframes, the degree of image impairment due to uncorrected errors depend on the BER. If the individual image impairments caused by channel errors are independent, then the overall effect is the summation of individual impairments. At higher error rates where separate errors cannot be considered independent anymore, we observe

a decreasing influence of the BER. We notice that in a bitstream, a sequence of L bits will be decoded erroneously if one of the bits is incorrect due to a channel error. The probability of any bit being decoded erroneously is then

$$P_{\mathrm{E}}(\mathrm{BER}, L) = 1 - (1 - \mathrm{BER})^{L}. \tag{18}$$

Note that this model describes the behavior related to dependencies between consecutive bits in the bitstream and does not assume any packetization. The value of L is therefore found by curve-fitting and not by an analysis of the data stream structure. Clearly, the value of L will be influenced by the implementation specifics such as resync markers. We interpret L as a value for the effective packet length, that is, the amount of data is lost after a single bit error as if an entire data packet of length L is lost due to an uncorrected error. This model for P_{E} corresponds very well with the observed channel-induced distortion behavior, so we postulate

$$D_{\mathrm{e}} \sim P_{\mathrm{E}} = \left(1 - (1 - \mathrm{BER})^{L}\right), \tag{19}$$

where parameter L was typically found to be in the order of 200 for intraframes and of 1000 for interframes.

(2) For intraframes, the degree of image impairment due to uncorrected errors does not only highly depend on the amount of variance of the original signal but also on the amount of quantization distortion. The expression $\mathrm{VAR}[X_i] - D_{\mathrm{q}}(r_i)$ represents the amount of variance that is encoded; the higher the distortion $D_{\mathrm{q}}(r_i)$, the less information is encoded. We observe that if $D_{\mathrm{q}}(r_i)$ increases, the effect of residual channel errors decreases. Clearly, at $r_i = 0$, nothing is encoded in this frame and the distortion equals the variance. At $r_i \gg 0$, $D_{\mathrm{q}} \approx 0$, there is no quantization distortion, all information is encoded and will be susceptible to bit errors. We therefore postulate

$$D_{\mathrm{e}}(r_i, \mathrm{BER}) \sim \mathrm{VAR}\left[X_i\right] - D_{\mathrm{q}}(r_i). \tag{20}$$

(3) For interframes, we did not observe a statistically significant correlation between the quantization distortion (i.e., the bit rate) and the image impairment due to channel errors. We assume that the image impairment is only related to the variance of the frame difference, thus, here we do not take into account the quantization distortion:

$$D_{\mathrm{e}}(r_i, \mathrm{BER}) \sim \mathrm{VAR}\left[X_i - X_{i-1}\right]. \tag{21}$$

These empirical observations lead us to postulate the following aggregated model of the channel-induced distortions for an intraframe:

$$\begin{aligned} D_{\mathrm{e}}(r_i, \mathrm{BER}) &= \mathrm{VAR}[\tilde{X}_i - \overline{X}_i] \\ &= \alpha P_{\mathrm{E}}(\mathrm{BER}, L_{\mathrm{I}})\left(\mathrm{VAR}\left[X_i\right] - D_{\mathrm{q}}(r_i)\right), \end{aligned} \tag{22}$$

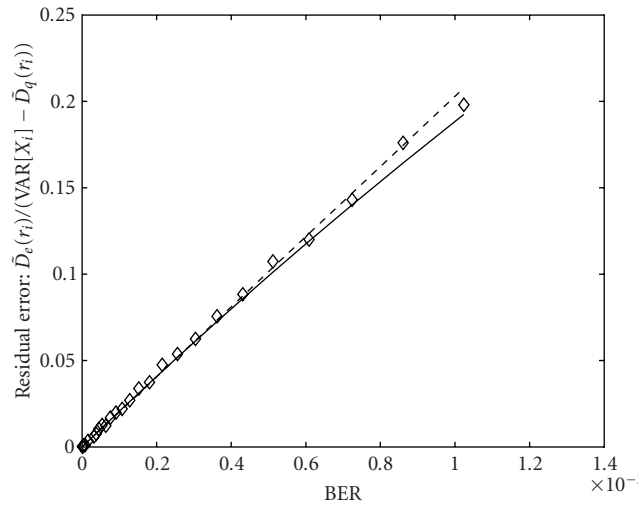

Figure 8: Plot of the normalized distortion $\tilde{D}_{\mathrm{e}}(r_i)/(\mathrm{VAR}[X_i] - \tilde{D}_{\mathrm{q}}(r_i))$ versus BER for the first intraframe of Carphone (shown by $\diamond$). The dashed line corresponds to the simple model $P_{\mathrm{E}} = \mathrm{BER}$ with $\alpha = 255.2$; the solid line to the model $P_{\mathrm{E}} = 1 - (1 - \mathrm{BER})^{202}$ with $\alpha = 1.29$.

and for one interframe:

$$D_{\mathrm{e}}(r_i, \mathrm{BER}) = \beta P_{\mathrm{E}}(\mathrm{BER}, L_{\mathrm{P}})\,\mathrm{VAR}\left[X_i - X_{i-1}\right]. \tag{23}$$

Here, $P_{\mathrm{E}}(\mathrm{BER}, L)$ is given by (18) and L_{I} and L_{P} are the effective packet lengths for intraframes and interframes, respectively. The constants α and β determine to which extent an introduced bit error distorts the picture and need to be found empirically.

For intraframes, $D_{\mathrm{e}}(r_i, \mathrm{BER})$ depends on BER and on the variance $\mathrm{VAR}[X_i] - D_{\mathrm{q}}(r_i)$. Two figures show the curve fitting on this two-dimensional function. Both figures show the results of encoding one frame at different bit rates (ranging from 0.05 to 2.0 bpp) and at different BERs (ranging from 10^{-3} to 10^{-6}), where bit errors were injected in the encoded bitstream randomly. Since we wish to predict the average behavior, we calculated the average distortions of 1000 runs for each setting as follows.

(1) Figure 8 shows the average $\tilde{D}_{\mathrm{e}}$ divided by $\mathrm{VAR}[X_i] - \tilde{D}_{\mathrm{q}}$ as a function of BER. The dashed line corresponds to a line fitted with $P_{\mathrm{E}} = \mathrm{BER}$ and $\alpha = 255.2$. We observe that it deviates at higher BER. The solid line corresponds to $P_{\mathrm{E}} = 1 - (1 - \mathrm{BER})^{L_{\mathrm{I}}}$ with an effective packet length $L_{\mathrm{I}} = 202$ and $\alpha = 1.29$, which gives a better fit.

(2) Figure 9 shows $\tilde{D}_{\mathrm{e}}$ divided by $P_{\mathrm{E}}(\mathrm{BER}, L_{\mathrm{I}} = 202)$ as a function of $\mathrm{VAR}[X_i] - \tilde{D}_{\mathrm{q}}$. The fitted line crosses the origin. Clearly, this model does not fit these measurements extremely well because the effect of $D_{\mathrm{q}}(r_i)$ is very unpredictable. On the other hand, because the model catches the coarse behavior, we still can incorporate the effect that $D_{\mathrm{q}}(r_i)$ has on the channel-induced distortion. For other sources (Foreman, Susie), we observe a similar behavior.

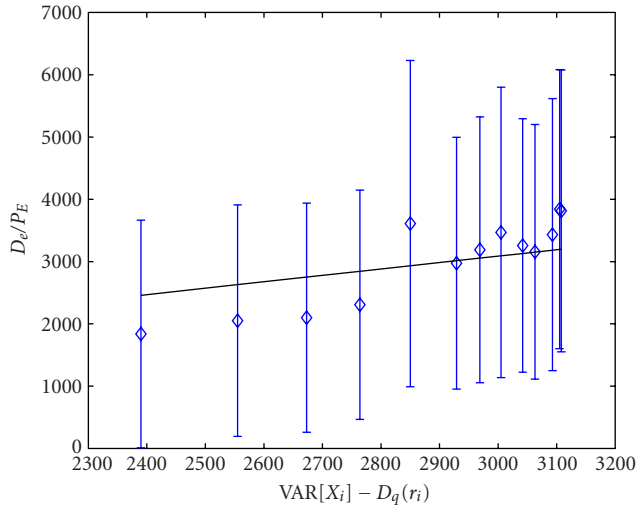

FIGURE 9: Plot of $\mathrm{VAR}[X_i] - \tilde{D}_q(r_i)$ versus the normalized distortion $\tilde{D}_e(r_i, \mathrm{BER})/P_E(\mathrm{BER}, L_I)$ (shown by $\diamond$) for the first intraframe of Carphone. The error bars represent the standard deviation over 1000 runs of the experiment. The solid line represents our model $D_e(r_i, \mathrm{BER})/P_E(\mathrm{BER}, L_I) = \alpha(\mathrm{VAR}[X_i] - D_q(r_i))$.

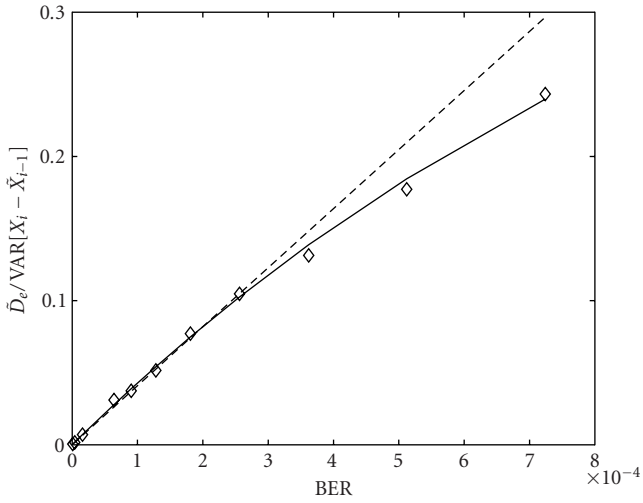

FIGURE 10: Plot of the normalized channel-induced distortion $\tilde{D}_e(r_i, \mathrm{BER})/\mathrm{VAR}[X_i - X_{i-1}]$ versus BER (shown by $\diamond$). The values are averaged over the first ten interframes of Carphone. The dashed line corresponds to the model $P_E = \mathrm{BER}$, and the solid line corresponds to the model $P_E = 1 - (1 - \mathrm{BER})^{876}$ with $\alpha = 0.51$.

Finally, for interframes, $D_e(r_i, \mathrm{BER})$ only depends on BER and on the constant factor $\mathrm{VAR}[X_i - X_{i-1}]$. Figure 10 shows the average $\tilde{D}_e$ divided by $\mathrm{VAR}[X_i - X_{i-1}]$ versus the BER. The resulting curve corresponds to $P_E = 1 - (1 - \mathrm{BER})^{L_P}$ with $L_P = 876$. Here, we found $\beta = 0.51$.

4.3.1. Error propagation in interframes

Due to the recursive structure of the interframe coder, decoding errors introduced in a frame will cause temporal error propagation [9, 14]. Since (5) and (6) tries to minimize the distortion over a whole GOP, we have to take this propagation into account for each frame individually. In [9], a high-level model was proposed to describe the error propagation in motion-compensated DCT-based video encoders including a loop filter. We adopted the λ factor which describes an exponential decay of the propagated error, but we discarded the γ factor which models propagation of errors in motion-compensated video, yielding

$$D_e(r_i, \mathrm{BER}) = (1-\lambda) D_e(r_{i-1}, \mathrm{BER}) + \beta(1 - (1 - \mathrm{BER})^{L_P})\,\mathrm{VAR}[X_i - X_{i-1}]. \tag{24}$$

Our observations are that this is an accurate model although the propagated errors decay only slightly. For instance, for the Carphone sequence, we found that $\lambda = 0.02$ (not shown here). In a coder where loop filtering is used to combat error propagation, this factor is much higher [9].

5. MODEL VALIDATION

We have now defined all models needed to solve (5) and (6). Assuming we know the variances $\mathrm{VAR}[X_i]$, $\mathrm{VAR}[X_i - X_{i-1}]$, the parameters for the functions $f(r)$, $g(r)$, and the model parameters κ, L_I, L_P, α, and β, we can minimize (5) and (6) using these models. Note that since in principle each frame can have its own RD function, the function will get the additional parameter i to signify that

$$\begin{aligned}
&D_{\mathrm{GOP}} = \frac{1}{N}\sum_{i=0}^{N-1}\{D_q(r_i|i) + D_e(r_i, \mathrm{BER}\,|i)\},\\
&D_q(r_0|i=0) = \mathrm{VAR}[X_0]\,2^{f(r_0|0)}, \quad \text{for } i = 0,\\
&D_e(r_0, \mathrm{BER}\,|i=0)\\
&\quad = \alpha(1-(1-\mathrm{BER})^{L_I})(\mathrm{VAR}[X_0] - D_q(r_0|0)), \quad \text{for } i > 0,\\
&D_q(r_i|i) = (\mathrm{VAR}[X_i - X_{i-1}]\\
&\qquad + \kappa D_q(r_{i-1}|i-1))\,2^{g(r_i|i)}, \quad \text{for } i > 0,\\
&D_e(r_i, \mathrm{BER}\,|i) = (1-\lambda) D_e(r_{i-1}, \mathrm{BER}\,|i)\\
&\qquad + \beta(1-(1-\mathrm{BER})^{L_P})\,\mathrm{VAR}[X_i - X_{i-1}],\\
&\qquad\qquad \text{for } i > 0.
\end{aligned} \tag{25}$$

In this section, we will verify these models by encoding a sequence of frames with different bit rate allocations and compare the measured distortion and the predicted distortion. Furthermore, we will introduce bit errors in the bitstream and verify the prediction of the distortion under error prone channel conditions. As mentioned in the introduction, in this paper, we do not optimize (5) and (6) using the models (25)—as would be required in a real-time implementation. Instead, we aim to show that it is possible to predict the overall distortion for a GOP under a wide range of channel conditions. We will show that a setting for N and r_i optimized with our behavior models (25) indeed yields a solution that is close to the measured minimum.

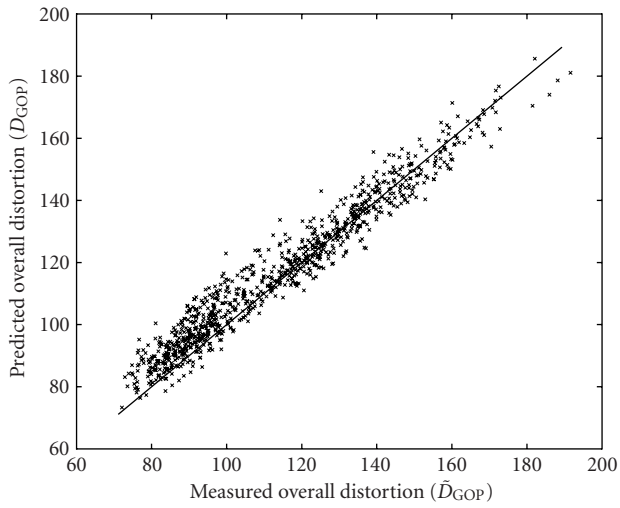

FIGURE 11: For each possible bit rate assignment, the cross (×) shows the measured distortion $\tilde{D}_{GOP}$ horizontally and the predicted distortion D_{GOP} vertically. The line represents the points where the measurements would match the predicted distortion.

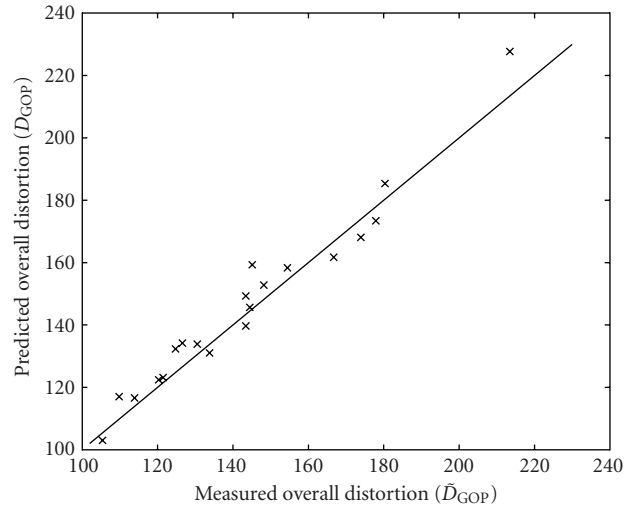

FIGURE 12: Selection of 20 bit rate assignments when BER = $32 \cdot 10^{-6}$. For each case the cross (×) shows the measured distortion $\tilde{D}_{GOP}$ horizontally and the predicted distortion D_{GOP} vertically. The solid line represents the points where the predicted distortion and the measured distortion would match.

To validate our model, we will compare the measurements of the overall distortion of a GOP with the predictions made with our model (25). We used the JPEG2000 encoder/decoder as our video coder (Figure 2), and encoded the Carphone sequence. In the first experiment, a GOP of ten frames was encoded with different bit rate allocations. No residual channel errors are introduced. In the second experiment, random bit errors were introduced in the encoded bitstream to simulate an error prone channel. In the third experiment, we addressed the issue of finding the optimal GOP length. In all these experiments, we used the models (25) and the parameters we have obtained in Section 4 for the first ten frames of Carphone. In the last experiment, we used our models to optimize the settings for a whole sequence. We compare optimizing the settings with our models and with two other simple rate allocations. Furthermore, we have investigated the gain that can be achieved if the RD curves are known for each individual frame instead of the average RD curves.

5.1. Optimal rate allocation

In this experiment, no residual channel errors were present (BER = 0) and the average bit rate available for each frame was 0.2 bpp. To each frame, we assigned bit rates varying from 0.1, 0.2, 0.3 to 1.1 bpp, while keeping the average bit rate constant at 0.2 bpp. The GOP length was set to 10. The total number of possible bit rate allocations with these constraints is 92378.

A GOP of ten frames was encoded with each of these bit rate allocations. We then measured the overall distortion denoted by $\tilde{D}_{GOP}$ and compared that with the predicted distortion D_{GOP} (using (4), (10), and (15)). Figure 11 shows the results. All points were plotted with the measured distortion $\tilde{D}_{GOP}$ on the horizontal axis. The vertical axis shows the predicted distortion D_{GOP}. The straight line corresponds to the points where the prediction matches the measured values. Points under this line underestimate the measured overall distortion and the points above the line overestimate the measured overall distortion. The region we are interested in is located in the lower left area where the bottom-most point represents the bit rate allocation that minimizes our model, D_{GOP} (25). The cloud shape gives good insight in the predictive strength of the model since the points are never far off the corresponding measured distortion.

As we can see in Figure 11, the predicted distortion and the measured distortion correspond well over the whole range of bit rate allocations. Note that although it is not possible with these proposed behavior models to find the exact values of r_i yielding the minimal measured distortion (we only know the exact distortion after encoding and decoding), the predicted minimal distortion is close to the measured minimum distortion. We use the following metrics to express the performance of the model: the relative error

$$\varepsilon_1 = \mathrm{E}\left[\frac{D_{GOP} - \tilde{D}_{GOP}}{\tilde{D}_{GOP}}\right] \cdot 100\%, \tag{26}$$

and the standard deviation of the relative error:

$$\varepsilon_2 = \mathrm{std}\left[\frac{D_{GOP} - \tilde{D}_{GOP}}{\tilde{D}_{GOP}}\right] \cdot 100\%. \tag{27}$$

For this experiment, $\varepsilon_1 = 3.2\%$, which means that we slightly overestimated all distortions; $\varepsilon_2 = 5.7\%$, which means that on average our predictions were within $3.2 - 5.7 = -2.5\%$ and $3.2 + 5.7 = 8.9\%$ around the measured values.

We can interpret this in terms of PSNR: an increase of the error variance of 5.7% corresponds to a decrease of the PSNR by $10 \log 1.089 = 0.37$ dB. This means that we predicted the average quality with 0.37 dB accuracy.

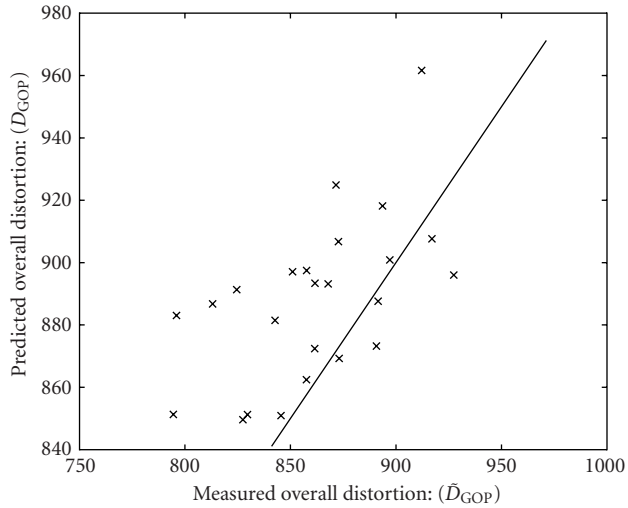

FIGURE 13: Selection of 20 bit rate assignments when BER = $1024 \cdot 10^{-6}$. For each case, the cross ($\times$) shows the measured distortion $\tilde{D}_{GOP}$ horizontally and the predicted distortion D_{GOP} vertically. The solid line represents the points where the predicted distortion and the measured distortion would match.

5.2. Optimal rate allocation for a channel with residual errors

When residual channel errors were introduced, the same experiment yielded different results at different runs because of the randomness of bit errors. Therefore, for each rate allocation, the coding should be done at least a thousand times and the measured distortion values should be averaged. Analyzing each bit allocation with such accuracy is very demanding in terms of computing time, therefore, we selected twenty cases uniformly distributed from the 92378 rate allocations to gain sufficient insight in the predictive power of the behavior models.

For this experiment, we chose BER = $32 \cdot 10^{-6}$. Figure 12 shows the measured average distortion $\tilde{D}_{GOP}$ and the predicted distortion D_{GOP} for the 10-frame case. Now, the relative error is $\varepsilon_1 = 2.0\%$ and $\varepsilon_2 = 3.7\%$.

Note that in these simulations, we did not use any special settings of a specific video coder and we used no error concealment techniques other than the standard JPEG2000 error resilience. Because of the combination of wavelet transforms and progressive bit plane coding in JPEG2000, in most cases the bit errors only caused minor distortions in the higher spatial frequencies. However, sometimes a lower spatial frequency coefficient was destroyed yielding a higher distortion.

Any individual random distortion can differ greatly from the predicted one. Because large distortions are less likely to occur than small distortions, our model gives a boundary on the resulting distortion. We measured that for 88.0% of the cases, the measured distortion was lower than the predicted value.

We then changed our BER to $1024 \cdot 10^{-6}$. Figure 13 shows the measured and the predicted distortions. For this high BER, the relative performance metrics were still good, $\varepsilon_1 = 0.31\%$ and $\varepsilon_2 = 3.6\%$. Note that these relative metrics are similar to the case without channel errors. This means that on the average, although the channel-error distortion is hard to predict, our model is still able to make good predictions of the average distortion even under error prone conditions. Apparently, the average D_e part of the total distortion is very predictable, this is probably due to the good error resilience of the JPEG2000 encoder we used.

5.3. Selection of the optimal GOP length

In the previous experiments, the optimal bit rate allocation was selected for each frame. This experiment deals with selecting the optimal GOP length N. The same constraints were used as in the previous experiment, but now the GOP length varied from 1 to 10.

Figure 14 shows for each GOP length from 1 to 10 the bit rate allocations for BER = 0. Observe that the average bit rate of 0.2 bpp per frame is spread out over each frame in the GOP to obtain a minimal overall distortion D_{GOP}. The last case ($N = 10$) corresponds to the bottom-most point in Figure 11.

Figure 15 shows the predicted overall distortion D_{GOP} and measured overall distortion $\tilde{D}_{GOP}$ for each of these bit rate allocations. Following our criterion (5) and (6), the optimal GOP length is $N = 8$. Since interframes are used, we expect that using larger GOPs gives lower distortions. This is generally true, but in these experiments we did not cover the whole solution space since we used increments of 0.1 bpp for the bit rates. With this limited resolution, we may find suboptimal solutions.

Figure 16 shows the result of a simulation where N varied from 1 to 15. In this simulation we only used our models to predict the distortion; the corresponding measurements were not carried out due to computational limitations (there are 600 000 combinations of rate allocations when bit rates $r_i \in \{0.1, 0.2, \ldots, 1.6\}$ are used). The distortions were again minimized with an average bit rate constraint of 0.2 bpp. The points correspond to the minimum achievable distortion D_{GOP} at each GOP length. We see that for $N > 6$, the average distortion did not substantially decrease anymore, so larger GOP lengths would not improve the quality greatly. Figure 16 also shows the results of the simulations for BER = $\{32 \cdot 10^{-6}, 256 \cdot 10^{-6}, 512 \cdot 10^{-6}\}$. Note that at some point, the accumulated channel-induced distortion becomes higher than the gain we obtain from adding another interframe. At this point, the internal controller should decide to encode a new intraframe to stop the error propagation.

5.4. Optimal rate allocation for whole sequences

In this experiment, we used our models and our optimization criterion to optimize the settings for the whole sequence of Carphone.

We have compared the measured distortion with two other simple rate allocation methods.

(1) The rates and GOP length settings are obtained using our models and optimization criterion with the

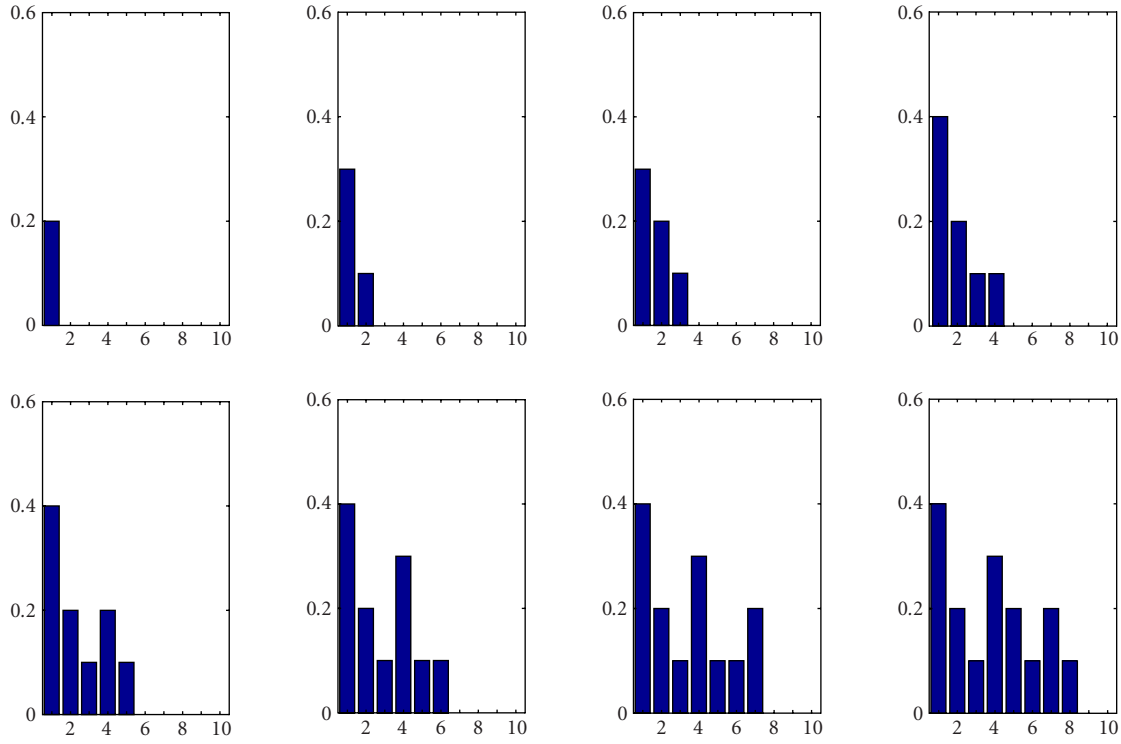

Figure 14: Bit rate allocations for BER = 0. Every plot corresponds to a GOP length running from $N = 1$ to 10. Within each plot, for each frame, the bit rate allocation that minimizes D_{GOP} is shown and the average bit rate of r is 0.2 bpp.

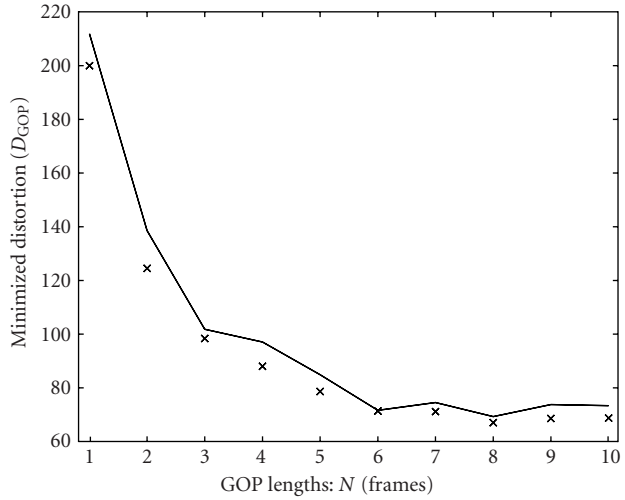

Figure 15: Minimized distortion D_{GOP} (—) and $\tilde{D}_{\text{GOP}}$ (×) for GOP lengths between 1 and 10 and for an average bit rate of $r = 0.2$ bpp.

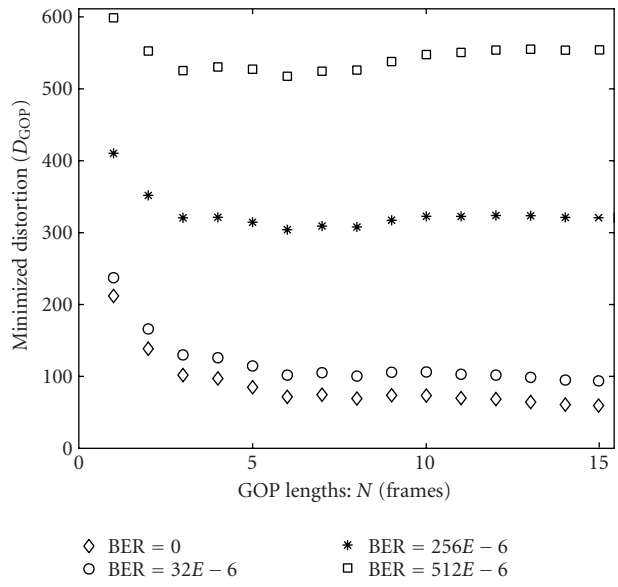

Figure 16: Minimized distortion D_{GOP} for GOP lengths between 1 and 15, for different BERs, and an average bitrate of $r = 0.2$ bpp.

constraints that $N_{\max} = 10$ and the average bit rate is 0.2.

(2) Every frame has the same fixed bit rate $r = 0.2$. The GOP length is obtained using our models and optimization criterion.

(3) Every frame has the same fixed bit rate $r = 0.2$. The GOP length has a fixed value of 10.

These methods were applied to the Carphone and the Susie sequences for BER = 0, BER = $128 \cdot 10^{-6}$, and BER = $512 \cdot 10^{-6}$. The results are shown in Table 1. For Carphone, method (1) is clearly better than method (3). Method (2) and

Table 1: Comparison between different rate allocation methods.

Case	Method	Distortion		
		1	2	3
Carphone	BER = 0	76.6	91.1	90.6
Carphone	BER = $128 \cdot 10^{-6}$	136.8	161.3	161.1
Carphone	BER = $512 \cdot 10^{-6}$	397.7	408.4	410.4
Susie	BER = 0	28.6	28.9	28.9
Susie	BER = $128 \cdot 10^{-6}$	47.4	49.6	59.5
Susie	BER = $512 \cdot 10^{-6}$	116.4	117.1	151.2

method (3) perform more or less the same. When bit errors are introduced, method (1) still outperforms the other two. For Susie, method (1) also outperforms the other two. When bit errors are present, method (2) (just adapting the GOP length) greatly outperforms method (3). We conclude that the performance of our method depends heavily on whether the characteristics of the source are changing over time or not. It seems that either optimizing the GOP length or the bit rates decreases the distortion as opposed to method (3).

Finally, we have investigated whether using RD parameters for each individual frame instead of average RD parameters, indeed gives a significant increase of the performance. We compared the case where for each individual frame the corresponding RD function is used for optimization (case 1), and the case where one average RD function is used for the whole sequence (case 2). For Carphone, we measured the following: for case 1, the average distortion $D = 76.5$, for case 2, $D = 91.0$. This means that significant gains can be expected when the RD curves are known for each frame. Of course in practice this is not possible. On the other hand, since consecutive frames look alike, we believe that an adaptive method to obtain the RD curves from previous frames could give significant gains. For Susie we have similar results. For case 1, $D = 28.6$, and for case 2, $D = 47.9$.

6. DISCUSSION

In this paper, we introduced a behavior model that predicts the overall distortion of a group of pictures. It incorporates the structure and prediction scheme of most video coders to predict the overall distortion on a frame-per-frame basis. Furthermore, the model corrects for statistical dependencies between successive frames. Finally, our model provides a way to predict the channel-induced distortion when residual channel errors are present in the transmitted bit steam.

Although the deviation of the model predicted distortion from the measured distortion can become substantial, with this model we can still compare different settings and select one likely to cause the smallest distortion.

Our models are designed to closely follow the behavior of the encoder, given the characteristics of the video data, and to make an accurate prediction of the distortion for each frame. These predictions are made before the actual encoding of the entire group of pictures. To predict the average distortion, we need to know the variance of each frame and the variance of the frame difference of the consecutive original frames. We also need two parameterized rate distortion curves and six other parameters (κ, α, β, L_I, L_P, and λ).

In our experiments—some of which were shown in this paper—we noticed that these parameters do not change greatly between consecutive group of pictures, therefore they can be predicted recursively from the previous frames that have already been encoded. On the other hand, we have shown that significant gains can be expected when the rate distortion parameters are obtained adaptively and no average rate distortion curves are used. The factors κ, α, β, L_I, L_P, and λ do not depend greatly on the source data, but rather on the coder design, and thus may be fixed for a given video encoder.

After obtaining the frame differences, the distortion can be predicted before the actual encoding takes place. This makes the model suitable for rate control and constant bit rate coding as well as for quality of service controlled encoders. Although this paper focused on rate allocation of entire frames rather than on macroblocks, all models can be generalized for use at the macroblock level.

REFERENCES

[1] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, International Thompson Publishing, London, UK, 1996.

[2] G. Côté, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error prone networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 952–968, 2000.

[3] G. M. Davis and J. M. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," in *Proc. SPIE Conference on Wavelet Applications of Digital Image Processing XIX*, vol. 2847, pp. 376–387, Denver, USA, 1996.

[4] M. Brystrom and J. W. Modestino, "Combined source channel coding for transmission of video over a slow-fading rician channel," in *Proc. International Conference on Image Processing*, vol. 2, pp. 147–151, Chicago, Ill, 1998.

[5] K. Stuhlmüller, N. Färber, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.

[6] A. van der Schaaf and R. L. Lagendijk, "Independence of source and channel coding for progressive image and video data in mobile communications," in *Proc. Visual Communications and Image Processing*, vol. 4067, pp. 187–197, Perth, Australia, June 2000.

[7] H. van Dijk, K. Langendoen, and H. Sips, "ARC: a bottom-up approach to negotiated QoS," in *Proc. 3rd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 128–137, Monterey, Calif, USA, December 2000.

[8] Y. S. Chan and J. W. Modestino, "Transport of scalable video over CDMA wireless networks: a joint source coding and power control approach," in *Proc. International Conference on Image Processing*, vol. 2, pp. 973–976, Thessaloniki, Greece, October 2001.

[9] N. Färber, K. Stuhlmüller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *Proc. International Conference on Image Processing*, vol. 2, pp. 550–554, Kobe, Japan, 1999.

[10] J. R. Taal, K. Langendoen, A. van der Schaaf, H. W. van Dijk, and R. L. Lagendijk, "Adaptive end-to-end optimization of mobile video streaming using QoS negotiation," in *Proc. International Symposium on Circuits and Systems*, vol. 1, pp. 53–56, Scottsdale, Ariz, USA, May 2002.
[11] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 533–545, 1994.
[12] M. Boliek et al., "Jpeg 2000 part 1 final committee draft, version 1.0," Tech. Rep., JPEG, 2002.
[13] G. Reyes, A. R. Reibman, and S. F. Chang, "A corruption model for motion compensated video subjected to bit errors," in *Proc. Packet Video Workshop '99*, NY, USA, April 1999.
[14] J. G. Kim, J. Kim, and C. C. J. Kuo, "Corruption model of loss propagation for relative prioritized packet video," in *Proc. SPIE Applications of Digital Image Processing XXIII*, vol. 4115, pp. 214–224, San Diego, July 2000.

Jacco R. Taal received his M.S. degree in Electrical Engineering from Delft University of Technology, Delft, The Netherlands, in 2001. At present he is pursuing his Ph.D. degree at the same university. His research interests include real-time video compression for wireless communications and peer-to-peer systems. Currently he is doing research on video transmissions for peer-to-peer communications and ad hoc networks.

Zhibo Chen received his B.S., M.S., and Ph.D. degrees from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 1998, 2000, and 2003, respectively. He is currently with Sony Research Center, Tokyo. His research interests include video coding theory and algorithm and video communication over networks.

Yun He received the B.S. degree in signal processing from Harbin Shipbuilding Institute, Harbin, China, in 1982, the M.S. degree in ultrasonic signal processing from Shanghai Jiaotong University, Shanghai, China in 1984, and the Ph.D. degree in image processing from Liege University, Liege, Belgium, in 1989. She is currently an Associate Professor at Tsinghua University, Beijing, China. She serves as a Senior Member in IEEE, Technical Committee Member of Visual Signal Processing and Communications in IEEE CAS Society, Picture Coding Symposium Steering Committee Member, as a Program Committee Member in SPIE Conference of Visual Communications and Image Processing (2000-2001). Her research interests include picture coding theory and methodology, picture coding algorithm software and hardware complexity analysis, video codec VLSI structure, and multiview and 3D picture coding.

R. (Inald) L. Lagendijk received his M.S. and Ph.D. degrees in electrical engineering from Delft University of Technology in 1985 and 1990, respectively. Since 1999, he has been a Full Professor in the Information and Communication Theory Group of Delft University of Technology. Prof. Lagendijk was a visiting scientist at Eastman Kodak Research (Rochester, NY) in 1991 and a Visiting Professor at Microsoft Research and Tsinghua University, Beijing, China, in 2000 and 2003. Prof. Lagendijk is the author of the book *Iterative Identification and Restoration of Images* (Kluwer, 1991) and coauthor of the books *Motion Analysis and Image Sequence Processing* (Kluwer, 1993) and *Image and Video Databases: Restoration, Watermarking, and Retrieval* (Elsevier, 2000). He has served as an Associate Editor of the *IEEE Transactions on Image Processing*, and he is currently an Associate Editor of the IEEE Transactions on Signal Processing Supplement on Secure Digital Media, and an Area Editor of Eurasip journal Signal Processing: Image Communication. At present his research interests include signal processing and communication theory, with emphasis on visual communications, compression, analysis, searching, and watermarking of image sequences. He is currently leading and actively involved in a number of projects in the field of data hiding and compression for multimedia communications.